

# Anforderung und Test: Ein ungleiches Paar, glücklich vereint

Vortrags-Handout, REConf 2008, München, 12. März 2008

Dr. Andreas Birk

Software.Process.Management, Gutenbergstraße 99, 70197 Stuttgart, andreas.birk@swpm.de

## Anforderungen und Testen miteinander verzahnen

Vielfach wird beim Zusammenspiel von Anforderungen und Testen nur die Ableitung von Testfällen aus Anforderungen betrachtet. Aber das greift viel zu kurz. Erst wenn Software-Organisationen einige weitere Facetten im Zusammenspiel von Anforderungen und Testen beachten, können sie Produktqualität und Produktivität der Entwicklung nachhaltig steigern.

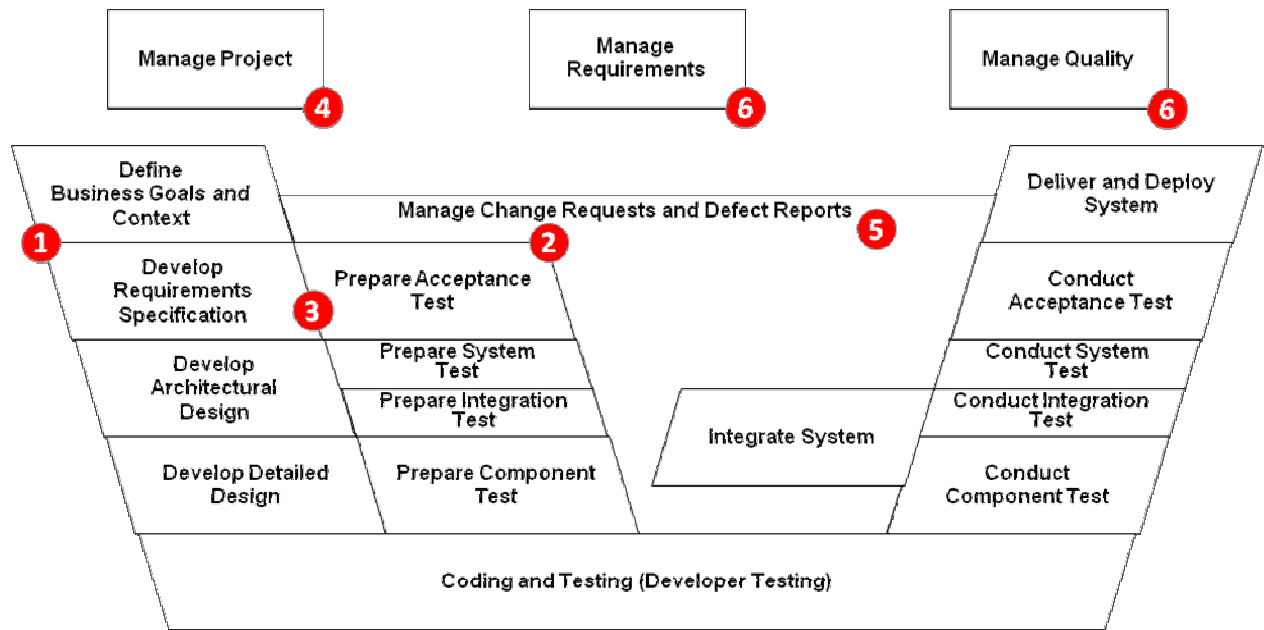
- Requirements-Management und Testen müssen über den gesamten Software-Lebenszyklus hinweg miteinander verzahnt sein
- Das steigert Produktqualität und die Produktivität der Entwicklung
- Der Weg dahin erfolgt schrittweise

## Anforderungen und Testen im Software-Lebenszyklus

Betrachtet man die Phasen des traditionellen Wasserfallmodells für die Software-Entwicklung, dann liegen Anforderungen und Testen weit auseinander. Die Anforderungen bilden den Startpunkt der Software-Entwicklung und die Grundlage, auf der fast alle weiteren Entwicklungstätigkeiten basieren. Das Testen bildet den Abschluss der Entwicklung, in dem sich zeigt, wie gut die vorangegangenen Phasen durchgeführt wurden.

In Wirklichkeit stehen Anforderungen und Testen auf vielfältige Weise eng miteinander in Beziehung. Die folgende Abbildung zeigt die Phasen der Software-Entwicklung in Anlehnung an das V-Modell. Sie betont die Aktivitäten der Testplanung und –vorbereitung (z.B. *Prepare Acceptance Test*), die möglichst direkt auf die ersten Entwicklungsaktivitäten folgen sollten (z.B. *Develop Requirements Specification*). Die Ziffern kennzeichnen Punkte, an denen ein besonders enger Zusammenhang zwischen Anforderungen und Testen besteht:

- Testbare Anforderungen (1)
- Anforderungsbasiertes Testen (2)
- Frühzeitige Testplanung und –vorbereitung (3)
- Anforderungsbasierte Testplanung und -priorisierung (4)
- Integriertes Anforderungs- und Fehlermanagement (5)
- Organisationsstrukturen und Rollen für Anforderungen und Testen (6)



## Testbare Anforderungen

Testbare Anforderungen besitzen die folgenden Eigenschaften:

- Granular & modular
- Präzise & prägnant
- Prüfkriterien (*Fit Criteria*)
- Kategorien von Anforderungen

Die ersten beiden Punkte fordern, dass die Anforderungen sozusagen „in kleinen Häppchen“ formell und stilistisch gut definiert sind—also nicht in umfangreichen, kaum strukturierten Prosadokumenten, und ohne Nominalisierungen, Passiv etc.

Jede Anforderung sollte Prüfkriterien enthalten (sog. *Fit Criteria*). Sie definieren, wie man später feststellt, ob eine Anforderung erfüllt ist. Außerdem sollte jede Anforderung einer oder mehreren Kategorien zu geordnet sein, entsprechend beispielsweise dem Volere-Template [RR06] oder IEEE Standard 830-1998 [IEE98]. Solche Kategorien machen eine Anforderungsspezifikation übersichtlich und erleichtern es, sie auf Vollständigkeit und Redundanzfreiheit zu überprüfen.

So sollten die Anforderung mittels Kategorien zunächst in verschiedene Ebenen eingeteilt sein:

- Geschäftsanforderungen (*Business Requirements*)
- Systemanforderungen

Die Geschäftsanforderungen spiegeln die Sicht der Auftraggeber wider. Jeder Geschäftsanforderung entsprechen eine oder mehrere Systemanforderungen, bei denen die Sicht der Software-Entwicklung im Vordergrund steht.

Anforderung und Test: Ein ungleiches Paar, glücklich vereint

Die Anforderungskategorien erleichtern auch den Übergang von den Anforderungen zur Testplanung und –vorbereitung. Denn meist werden die Anforderungen einer Kategorie auf eine bestimmte Art getestet. Die folgenden Anforderungsarten sind für das Testen von besonderer Bedeutung. Sie korrespondieren mit den vier wesentlichen Testarten:

- Funktionale Anforderungen (bzgl. funktionale Tests)
- Performance-Anforderungen (bzgl. Last- und Performance-Tests)
- Benutzbarkeitsanforderungen (bzgl. Usability Tests)
- Sicherheitsanforderungen (bzgl. Application Security Tests)

Zwei Anforderungskategorien zielen speziell auf das Testen ab:

- *Test-Projektanforderungen*: Projektanforderungen hinsichtlich der Testdurchführung. Dazu zählen Anforderungen über die zeitliche Verfügbarkeit von Systemteilen oder Festlegungen (z.B. Schnittstellen- oder Datenformatsdefinitionen), so dass eine systematische Testvorbereitung und –durchführung möglich ist.
- *Testbarkeitsanforderungen*: Funktionale Anforderungen an das System, die nur dazu dienen, dass die erforderlichen Tests durchgeführt werden können (z.B. besondere Schnittstellenfunktionen, die nur für Tests benötigt werden).

Auch die Betriebsanforderungen nehmen eine besondere Rolle hinsichtlich des Testens ein. Denn viele Tests werden während des Systembetriebs durchgeführt (z.B. Installationserfolgstests oder Last- und Performance-Tests).

## Anforderungsbasiertes Testen

Anforderungsbasiertes Testen setzt voraus, dass zu den Testfällen die zugrunde liegenden Anforderungen definiert und vermerkt sind (Verfolgbarkeit, *Traceability*). Das ist der Fall, wenn die Testfälle systematisch aus den Anforderungen abgeleitet werden. Häufig aber existieren Tests, zu denen die Anforderungen nicht ausreichend dokumentiert oder zumindest nicht explizit vermerkt sind. – Wie kann eine Software-Organisation in dieser Situation die Vorteile des anforderungsbasierten Testens schnell nutzen?

Viele Vorteile des anforderungsbasierten Testens erschließen sich bereits, wenn die Testfälle nach Anforderungskategorien gruppiert sind. Dies ermöglicht zum Beispiel risikobasiertes Testen und Monitoring der Testdurchführung nach Geschäftsprozessen. Eine ausführliche Definition der betreffenden Anforderungen kann später schrittweise ergänzt werden.

Dazu werden Anforderungskategorien benötigt, die die existierenden Testfälle strukturieren. Dies sind im Grunde die gleichen Kategorien wie oben unter *Testbare Anforderungen* beschrieben. In vielen Fällen reichen für die Strukturierung existierender Tests jedoch zunächst die folgenden drei Kategorien:

- *Geschäftsprozesse*: Eine Hierarchie von Geschäftsprozessen, in denen die zu testenden Systeme eingesetzt werden. Diese Hierarchie dient dazu, die Testfälle aus Sicht der Geschäftsanforderungen (*Business Requirements*) zu strukturieren. Beispielsweise können die Tests so nach Relevanz aus Sicht der Auftraggeber gruppiert und priorisiert werden.

Anforderung und Test: Ein ungleiches Paar, glücklich vereint

- *Systemstruktur*: Eine Hierarchie der Systemaggregation aus den zu testenden Gesamtsystemen, ihren Teilsystemen und Komponenten. Diese Hierarchie dient dazu, Tests nach zu testenden Systemeinheiten auszuwählen, zum Beispiel für Regressionstests nach Änderungen.
- *Anforderungsarten*: Eine Hierarchie aus wesentlichen funktionalen und nichtfunktionalen Anforderungsarten sowie deren Unterkategorien—z.B. Funktionalitätsgruppen und nicht-funktionale Anforderungen wie Performance- und Sicherheitsanforderungen. Diese Hierarchie dient dazu, Testfälle zu bestimmten Aspekten schnell auszuwählen und fokussiert zu testen (z.B. die Benutzerverwaltung oder das Performance-Verhalten).

## Frühzeitige Testplanung und -vorbereitung

Testen ist keineswegs die „letzte“ Entwicklungsphase, die erst nach Abschluss der Implementierung beginnt. Planung und Vorbereitung des Testens können und sollten bereits parallel zur Anforderungsspezifikation beginnen. Nur dann ist sicher gestellt, dass die Tests gut vorbereitet sind. Außerdem trägt die Definition der Testfälle dazu bei, die Anforderungen früh zu überprüfen und ihre Qualität zu sichern.

Zwei Maßnahmen bei der Projektplanung helfen, dass eine frühe Testvorbereitung und -planung gewährleistet ist:

- *Testen als (Teil-)Projekt*: Das Testmanagement von der Testplanung und -vorbereitung bis zur Testdurchführung und -auswertung sollte als ein eigenständiges Teilprojekt organisiert sein. So erhalten die Testaktivitäten den Stellenwert im Projekt, den sie benötigen. Auch das Zusammenspiel zwischen Anforderungen und Testen steht dann auf stabilen Füßen.
- *Quality Gates zur Verifikation von Testfällen und Anforderungen*: Testfälle sollten frühzeitig definiert und ihre Korrektheit gegenüber den Anforderungen in systematischen Reviews verifiziert werden. Diese Prüfungen sind auch eine erste Gelegenheit, um die Korrektheit der Anforderungen abzusichern. Werden derart frühzeitig Unklarheiten oder Unstimmigkeiten in den Anforderungen entdeckt, können sie noch ohne zu großen Aufwand korrigiert werden.

## Anforderungsbasierte Testplanung und -priorisierung

Tests müssen immer priorisiert und nach Maßgabe der verfügbaren Ressourcen durchgeführt werden. Ein sehr wichtiges Kriterium für die Priorisierung sind die geschäftlichen Risiken (d.h. Kritikalität und Eintrittswahrscheinlichkeit), die mit den unterschiedlichen Anforderungen verbunden sind. Diejenigen Anforderungen, die besonders geschäftskritisch sind, müssen vordringlich getestet werden.

Grundlagen für eine solche risikobasierte Testplanung sind die oben beschriebenen Anforderungskategorien, insbesondere die Geschäftsanforderungen. Wenn sie mit den Testfällen verbunden sind, können die Testfälle gemäß der Geschäftskritikalität ausgewählt werden. Mithilfe der Geschäftsanforderungen lässt sich die Risikobewertung gemeinsam mit den Auftraggebern sehr viel leichter vornehmen, als es allein auf Grundlage der Testfälle möglich wäre.

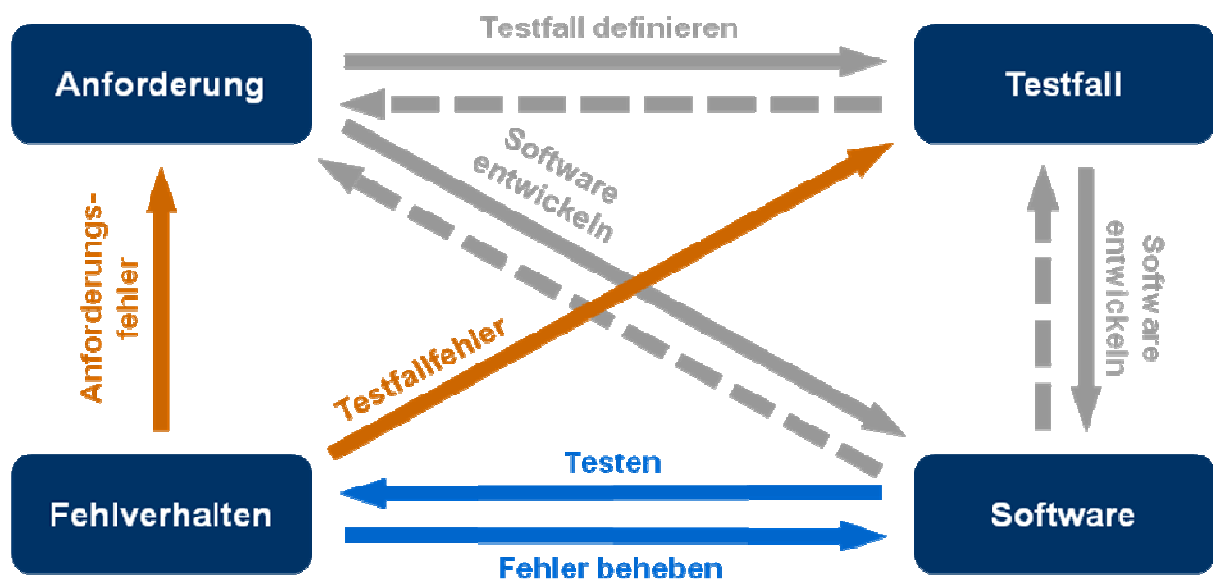
Anforderung und Test: Ein ungleiches Paar, glücklich vereint

## Integriertes Anforderungs- und Fehlermanagement

Das Fehlermanagement vermerkt die im Test identifizierten Fehler und plant und dokumentiert ihre Analyse und Behebung. Die Aktivitäten des Fehlermanagements ähneln sehr denen des kontinuierlichen Anforderungsmanagements. Außerdem kann identifiziertes Fehlverhalten dazu führen, dass neue Anforderungen definiert oder existierende Anforderungen geändert werden.

Aus diesen Gründen sollte in einer Software-Organisation ein zentrales, integriertes Anforderungs- und Fehlermanagement installiert sein. Es sollte sowohl die Software-Entwicklung umfassen wie auch IT-Betrieb, Benutzer der Fachbereiche sowie Support und Service-Desk.

Die Abbildung veranschaulicht, wie identifiziertes Fehlverhalten auf die Anforderungen zurück wirken und Änderungsaufträge auslösen kann. Identifiziertes Fehlverhalten kann prinzipiell drei Gründe haben: (1) Tatsächlicher Fehler, (2) falsche Angaben in den Testfällen (und somit kein wirkliches Fehlverhalten des getesteten Systems) und (3) fehlerhafte Anforderungen (z.B. aufgrund widersprüchlicher Anforderungen). Diese Rückkopplungen zu Tests und fehlerhaften Testfällen und Anforderungen lassen sich in einem übergreifenden Anforderungs- und Fehlermanagement am effizientesten steuern und überwachen.



## Organisationsstrukturen und Rollen für Anforderungen und Testen

In vielen Software-Organisationen ist das Requirements-Management keiner einzelnen Rolle oder organisatorischen Einheit zugeordnet. Vielmehr sind jeweils Teilaufgaben des Requirements-Managements auf mehrere Rollen verteilt, die eigentlich andere Aufgabenschwerpunkte besitzen: Produkt- und Portfoliomanagement, Software-Architektur und Projektleitung.

Testen und Testmanagement hingegen bilden meist klare Aufgabenschwerpunkte, die einem festen Personenkreis zugeordnet sind, in der Regel als Teil des Qualitätsmanagements. Diese

Anforderung und Test: Ein ungleiches Paar, glücklich vereint

Organisationsstruktur des Testens kann als Ausgangspunkt dienen, um auch das Requirements-Management zu zentralisieren und so die Qualität der Software und die Produktivität der Entwicklung weiter zu steigern.

Fallbeispiel: Die Software-Abteilung eines Finanzdienstleisters hat die Testaktivitäten schrittweise zentralisiert und damit auch die Grundlage für die Zentralisierung des Requirements-Managements geschaffen. Am Anfang wurde die Verantwortung für Last- und Performance-Tests aller Anwendungen in ein zentrales Testteam überführt. Danach übernahm dieses Team auch die Verwaltung der funktionalen Testfälle, die zuvor von mehreren Fachbereichen getrennt gepflegt wurden. Dadurch ist eine Infrastruktur etabliert worden, die nun auch für das zentrale Anforderungsmanagement und somit für anforderungsbasiertes Testen genutzt werden kann.

Oft leiden Qualität und Effizienz des Testens darunter, dass die Tester zu wenig über die fachlichen Anforderungen und die technische Realisierung des Systems bescheid wissen. Mögliche Maßnahmen dagegen sind Job-Rotation und enge Kommunikation und Zusammenarbeit zwischen den Testern und den anderen Rollen. Im Rahmen der Job-Rotation sollten Anforderungs- und Architektur-Experten zeitweise Test-Rollen besetzen. Zusammenarbeit und Kommunikation können durch Quality-Gates für Anforderungs- und Test-Dokumente gefördert werden, bei denen immer auch Requirements-Manager, Software-Architekten und Testmanager beteiligt sein müssen.

## Schrittweise Verzahnung von Anforderungen und Testen

Der Weg zur stärkeren Verzahnung von Anforderungen und Testen muss schrittweise erfolgen. Als erste Schritte und Zwischenstationen bieten sich die folgenden Stufen an:

1. Kategoriensystem zur Strukturierung von Anforderungen und Testfällen etablieren
2. Zu allen Anforderungen Prüfkriterien (*Fit Criteria*) definieren und Testfälle frühzeitig daraus ableiten
3. Testen als (Teil-)Projekt etablieren und Quality Gates für Testfälle einführen
4. Anforderungsbasierte Testplanung und -priorisierung aufbauen
5. Abläufe und Infrastruktur für Anforderungs- und Fehlermanagement integrieren

## Literatur

[IEE98] IEEE Recommended Practice for Software Requirements Specification, IEEE STD 830-1998, 1998

[RR06] Suzanne Robertson and James Robertson, Mastering the requirements process (2<sup>nd</sup> Ed.), Addison-Wesley Publishing Co., New York, NY, 2006